Self-Taught Metric Learning without Labels

Sungyeon Kim\textsuperscript{1} Dongwon Kim\textsuperscript{1} Minsu Cho\textsuperscript{1,2} Suha Kwak\textsuperscript{1,2}

Dept. of CSE, POSTECH\textsuperscript{1} Graduate School of AI, POSTECH\textsuperscript{2}

\{sungyeon.kim, kdwon, mscho, suha.kwak\}@postech.ac.kr

http://cvlab.postech.ac.kr/research/STML/

Abstract

We present a novel self-taught framework for unsupervised metric learning, which alternates between predicting class-equivalence relations between data through a moving average of an embedding model and learning the model with the predicted relations as pseudo labels. At the heart of our framework lies an algorithm that investigates contexts of data on the embedding space to predict their class-equivalence relations as pseudo labels. The algorithm enables efficient end-to-end training since it demands no off-the-shelf module for pseudo labeling. Also, the class-equivalence relations provide rich supervisory signals for learning an embedding space. On standard benchmarks for metric learning, it clearly outperforms existing unsupervised learning methods and sometimes even beats supervised learning models using the same backbone network. It is also applied to semi-supervised metric learning as a way of exploiting additional unlabeled data, and achieves the state of the art by boosting performance of supervised learning substantially.

1. Introduction

Understanding similarities between data is at the heart of many machine learning tasks such as data retrieval [29, 37, 47, 48], face verification [35, 45], person re-identification [7, 56], few-shot learning [41, 46, 49], and representation learning [29, 52, 61]. Metric learning embodies the perception of similarity by learning an embedding space where the distance between a pair of data represents their inverse semantic similarity. Also, the mapping from data to such a space is typically modeled by deep neural networks.

Recent advances in metric learning rely heavily on supervised learning using large-scale datasets. Manual annotation of such datasets is however costly, and thus could limit the class diversity of training data and in consequence the generalization capability of learned models. Unsupervised metric learning has been studied to resolve this issue. Existing methods in this line of research mainly synthesize class information of training data by assigning a surrogate class per training instance [13, 55, 59, 60] or discovering pseudo classes through $k$-means clustering [3, 4, 26, 33, 34, 57], hierarchical clustering [57], or random walk [25]. Although these methods have demonstrated impressive results without using groundtruth labels in training, they often fail to capture intra-class variation [13, 55, 59, 60] or impose substantial computational burden due to the off-the-shelf techniques [3, 4, 26, 33, 34, 57].

In this paper, we propose a new method for unsupervised metric learning that addresses the aforementioned limitations of previous work and achieves the state of the art as shown in Fig. 1. The major contribution of this paper is two-fold. First, we introduce a novel, end-to-end self-taught metric learning framework (STML), whose overall pipeline is illustrated in Fig. 2. Unlike the existing works, it predicts class-equivalence relations between data within each mini-batch by self-exploration and leverages the predicted relations as synthetic supervision for metric learning. Our framework manages two embedding networks, namely
teacher and student whose backbone are initialized identically. Class-equivalence between a pair of data is approximately estimated as their semantic similarity on the embedding space of the teacher network. The predicted similarity is used as a soft pseudo label for learning the student model, and the teacher model is in turn updated by a momentum-based moving average of the student. Iterating this process evolves the student model progressively.

The success of STML depends heavily on the quality of predicted semantic similarities, and our second contribution lies in the way of estimating semantic similarities using contexts. Specifically, given a pair of data, we compute their semantic similarity considering the overlap of their contexts (i.e., neighborhoods in the embedding space) as well as their pairwise distance. We found that the contextualized semantic similarity approximates class-equivalence precisely. Further, since it has a real value indicating the degree of semantic similarity, it provides rich information beyond binary class-equivalence. Also, since it requires no external module nor memory banks [20, 31], it makes STML efficient and concise. To further enhance the quality of predicted semantic similarity, we design the teacher network to learn a higher dimensional embedding space than the student counterpart. This asymmetric design of the two networks allows the teacher to provide more effective supervision thanks to its improved expression power while the student, i.e., our final model, remains compact.

To the best of our knowledge, STML is the only unsupervised metric learning method that can consider semantic relations between data in end-to-end training without introducing off-the-shelf techniques. Compared to the instance-level surrogate classes [13, 55, 59, 60], pseudo labels generated and exploited by STML are more appropriate to capture semantic relations between data since they indicate class-equivalence relations. Also, unlike previous work based on pseudo labeling [3, 4, 25, 26, 33, 57], STML employs no external algorithm and thus allows training to be efficient, end-to-end, and less sensitive to hyper-parameters. Further, it is naturally applied to semi-supervised metric learning [14] as well as unsupervised metric learning.

We first evaluate STML on standard benchmarks for metric learning [32, 48, 54], where it largely outperforms existing unsupervised learning methods. Surprisingly, sometimes it even beats some of supervised learning models using the same backbone network as shown in Fig. 1. Beyond that, its efficacy is demonstrated on two benchmarks for semi-supervised metric learning [14], where it substantially outperforms previous work as well.

2. Related Work

**Unsupervised metric learning** has been mainly addressed in two different directions, instance discrimination [13, 55, 59, 60] and pseudo labeling [3, 4, 25, 26, 33, 34, 57]. Following the contrastive learning strategy [2, 6, 8, 20], instance discrimination methods assign a unique label per training instance and learn embedding spaces where different instances are well discriminated. Unfortunately, they have a trouble modeling variations within each latent class. On the other hand, pseudo labeling methods discover pseudo classes by applying off-the-shelf algorithms like $k$-means clustering [3, 4, 26, 33, 34], hierarchical clustering [57] and random walk [25] to unlabeled training data. Such methods can consider the class-equivalence relations between training data by utilizing existing supervised metric learning losses [19, 45, 53] with pseudo labels. How-
ever, the major drawback of these methods is their prohibitive high complexity imposed by the auxiliary algorithms, which obstructs end-to-end training.

**Semi-supervised metric learning**, recently introduced by Duan et al. [14], assumes that a training dataset consists of both labeled and unlabeled data. The previous work discovers and assigns pseudo labels to unlabeled training data via k-means clustering. Unfortunately, it also suffers from the aforementioned drawbacks of the unsupervised learning using pseudo labeling.

**Self-supervised representation learning** aims at learning high-quality intermediate features to generalize to other downstream tasks. To this end, various annotation-free pretext tasks have been proposed, e.g., rotation prediction [17], solving jigsaw puzzle [38], image colorization [62], and image inpainting [40]. Pseudo labeling also has been widely used as it allows to learn models using conventional losses; existing methods in this direction discover pseudo classes by clustering [4] or treat each training instance as a surrogate class [6, 13, 20, 55]. Also, as an extension of the surrogate class, nearest neighbors have been leveraged as a positive set for self-supervised learning [15, 23, 31], but they often produce noisy labels unless a sufficiently trained embedding space is provided in advance.

### 3. Proposed Method

This section first presents an overview of our self-taught metric learning framework, dubbed STML. We then describe the each step of STML in detail and how to estimate the contextualized semantic similarity, which is the synthetic supervision for our framework.

#### 3.1. Overview

STML enables efficient end-to-end training without using any off-the-shelf technique, while capturing semantic relations between data by generating and exploiting pseudo labels that approximate their class-equivalence.

Fig. 2 illustrates the overall pipeline of STML. It manages two embedding networks with the identically initialized backbone: a teacher and a student. The teacher network is used to estimate semantic similarities between data as their approximate class-equivalence relations, which are in turn used as synthetic supervision for training the student network. In particular, the teacher model leverages a high-dimensional (e.g., 1024 or 2048) embedding layer $g^t$ with the backbone encoder $g^t$. The use of the 1024-dimensional space allows the teacher embedding layer to encode richer information and produce more reliable semantic similarity. Our student model has two parallel embedding layers $f^s$ and $g^s$ that share the backbone encoder $g^s$. $f^s$ is the embedding layer of our final model, and has a lower dimension (e.g., 128 or 512) to learn a compact embedding space. $g^s$ is an auxiliary layer dedicated to continually updating the teacher embedding layer $g^t$, and thus having the same output dimension with that of $g^t$.

Algorithm 1 describes the procedure of STML in detail. First, the teacher network estimates semantic similarities that approximate class-equivalence relations between data, which are in turn used as synthetic supervision for training the student network. Then the teacher network is updated by the momentum-based moving average of the student. Our final model is the student network obtained by iterating this alternating optimization a predefined number of times. In the next section, we will elaborate each step of STML.

#### 3.2. Self-Taught Metric Learning Framework

**Step #1: Nearest neighbor batch construction.** The way of constructing mini-batches for training has a massive impact on performance in metric learning [16, 44]. One of the most widely used methods for batch construction is the pk-sampling: It randomly chooses $p$ unique classes, and then randomly samples $k$ instances per class. Since this method leverages class labels of training data, it is however prohibited in an unsupervised learning setting. We instead propose a new strategy based on the nearest neighbor search. For each mini-batch, it randomly samples $q$ queries and then searches for $k - 1$ nearest neighbors of each query; including both queries and their nearest neighbors, the mini-batch becomes of size $qk$. Our strategy is simple, fast, yet allows mini-batches to include diverse and relevant samples.

**Step #2: Generating synthetic supervision.** Supervised metric learning methods in general employ as supervision class-equivalence relations between data, which are how-
ever not available in our setting. STML instead approximates the class-equivalence relations by semantic similarities between data on the teacher embedding space, which are used as synthetic soft supervision for learning the student model. The major challenge is that the teacher embedding space is not mature enough to grasp the semantic relations between data at early stages of unsupervised learning. We thus propose contextualized semantic similarity, which measures the similarity between a pair of data while considering their contexts on the teacher embedding space; its details are presented in Sec. 3.3.

**Step #3: Learning the student model.** We simultaneously optimize the two branches of the student model using the contextualized semantic similarities as synthetic supervision. Since its value is soft, however, the contextualized semantic similarity cannot be incorporated with the ordinary metric learning losses that take discrete labels as supervision. We thus employ the relaxed contrastive loss [28], which is designed to utilize soft relational labels for metric learning. Let \( w_{ij} \) be the contextualized semantic similarity between \( x_i \) and \( x_j \), and \( z_i^{f_s} \) denote the student embedding vector of \( x_i \) generated by \( f_s \). Defining \( d_{ij}^{f_s} := \frac{||z_i^{f_s} - z_j^{f_s}||_2}{(\frac{1}{n} \sum_{k=1}^{n} ||z_k^{f_s} - z_k^{f_s}||_2)} \) as the relative distance between \( f_s \) and \( f_s \), the loss is then given by

\[
\mathcal{L}_{RC}(Z^{f_s}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i} w_{ij} (d_{ij}^{f_s})^2 + \frac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i} (1 - w_{ij}) [\delta - d_{ij}^{f_s} + 1]_+ ,
\]

where \( Z^{f_s} \) is all student embedding vectors generated by \( f_s \), \( n \) is the number of samples in the batch, and \( \delta \) is a margin. The contextualized semantic similarity serves as the weights for the attracting and repelling terms of the loss, and thus determines the magnitude of the force that pulls or pushes a pair of data on the student embedding space. Note that \( g_s \) is trained in the same manner on \( f_s \). We additionally apply self-distillation to further utilize complex higher dimensional information for lower dimensional space. Following [43], we use Kullback-Leibler divergence as objective of self-distillation:

\[
\mathcal{L}_{KL}(Z^{f_s}, Z^{g_s}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \neq i} \psi(-d_{ij}^{g_s}) \log \left( \frac{\psi(-d_{ij}^{g_s})}{\psi(-d_{ij}^{f_s})} \right),
\]

where \( \psi(\cdot) \) is softmax operation. Note that the gradient-flow of \( g_s \) is truncated. In summary, STML trains the two branches \( f_s \) and \( g_s \) of the student model by minimizing the overall loss defined as

\[
\mathcal{L}_{STML}(Z^{f_s}, Z^{g_s}) = \frac{1}{2} [\mathcal{L}_{RC}(Z^{f_s}) + \mathcal{L}_{RC}(Z^{g_s})] + \mathcal{L}_{KL}(Z^{f_s}, Z^{g_s}).
\]

**Step #4: Momentum update of the teacher model.** The teacher model should be updated as the student model is trained so that the synthetic supervision computed from the teacher embedding space is enhanced progressively. The simplest update strategy is to replace the teacher model with the student counterpart at every iteration. However, this could deteriorate the consistency of the synthetic supervision over iterations, which leads to unstable training of the student model. We thus update the teacher model by the momentum-based moving average of the student, except for the embedding layer \( f_s \) since the teacher model has only a high-dimensional one. Let \( \theta_t \) be the parameters of the teacher model and \( \theta_t \) be the parameters of \( \phi_t \) and \( g_t \). Then \( \theta_t \) is updated by

\[
\theta_t \leftarrow m \theta_t + (1 - m) \theta_s,
\]

where \( m \in [0, 1] \) is the coefficient that controls the momentum update rate, slowing down the update of the teacher model as its value increases.

### 3.3. Contextualized Semantic Similarity

The quality of predicted semantic similarities is of vital importance in STML since they are the only supervision for training. However, it is not straightforward to compute reliable semantic similarities especially at early stages of unsupervised learning where the teacher network is significantly incomplete. To overcome this challenge, we propose to use context, i.e., neighborhood on a data manifold.

Our key idea is not to rely on the relations considering two images of interest only, but to figure out their indirect semantic relations via their contexts. Their multiple indirect relations can contribute to capturing common patterns frequently observed within each class and thus understanding the underlying class-equivalence relations. To this end, we design the contextualized semantic similarity. Given a pair of samples, their contextualized semantic similarity is defined as a combination of their pairwise similarity and contextual similarity on the teacher embedding space.

**Pairwise similarity.** The pairwise similarity of two samples \( x_i \) and \( x_j \), denoted by \( w_{ij}^{f_s} \), is given by

\[
w_{ij}^{f_s} = \exp \left( -\frac{||z_i^{f_s} - z_j^{f_s}||_2^2}{\sigma} \right),
\]

where \( z_i^{f_s} \) is the teacher embedding vector of \( x_i \) and \( \sigma \) is the Gaussian kernel bandwidth. This kind of similarity has been known to encode rich information about relations of \( x_i \) and \( x_j \) when the teacher embedding space is pre-trained in a fully supervised manner [28], but is often broken in unsupervised metric learning. We thus supplement it with the contextual similarity, which is detailed below.

**Contextual similarity.** The underlying assumption of the contextual similarity is that the more two samples semantically similar, the larger their contexts overlap. One may
consider nearest neighbors of a data point on the teacher embedding space as its context, but they often include irrelevant data when the embedding space is not sufficiently trained yet. We thus instead consider $k$-reciprocal nearest neighbors as context since they are known to discover highly correlated candidates [42, 63]. The set of $k$-reciprocal nearest neighbors of a data point $x_i$ is given by

$$R_k(i) = \{j | (j \in N_k(i)) \land (i \in N_k(j))\}, \quad (6)$$

where $N_k(i)$ is the $k$-nearest neighbors of $x_i$, including itself (i.e., $i \in N_k(i)$), in the current mini-batch. The initial form of contextual similarity measures the degree of overlap between $R_k(i)$ and $R_k(j)$ while considering whether $x_i$ and $x_j$ are in $k$-reciprocal nearest neighbor relation. Since the size of the neighbor set of $x_i$ and $x_j$ would be different, we design an asymmetric Jaccard similarity as

$$\tilde{w}_{ij}^C = \begin{cases} \frac{|R_k(i) \cap R_k(j)|}{|R_k(i)|}, & \text{if } j \in R_k(i), \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

In addition, we adopt the idea of query expansion [1, 10, 11] to further improve its reliability. Specifically, $\tilde{w}_{ij}^C$ is reformulated as the average of the contextual similarities between $x_j$ and nearest neighbors of $x_i$ as follows:

$$\bar{w}_{ij}^C = \frac{1}{|N_k^2(i)|} \sum_{h \in N_k^2(i)} \tilde{w}_{hj}^C. \quad (8)$$

The expanded version in Eq. (8) is not symmetric anymore. To ensure symmetry, the final form of the contextual similarity is defined as the average of $\bar{w}_{ij}^C$ and its transpose, which is given by

$$w_{ij}^C = w_{ji}^C = \frac{1}{2}(\bar{w}_{ij}^C + \bar{w}_{ji}^C). \quad (9)$$

Contextualized semantic similarity. Finally, the contextualized semantic similarity between $x_i$ and $x_j$ is defined as the average of their pairwise similarity in Eq. (5) and contextual similarity in Eq. (9):

$$w_{ij} = \frac{1}{2}(w_{ij}^P + w_{ij}^C). \quad (10)$$

Note that both of $w_{ij}^P$ and $w_{ij}^C$ lie within $[0, 1]$ thus their scales are balanced. The contextualized semantic similarity effectively represents semantic relatedness between data by considering both of their pairwise relation and context. We found that the contextualized semantic similarity is highly correlated with groundtruth class-equivalence, which is empirically verified in Sec. 4.1.4.

4. Experiments

This section validates the effectiveness of STML on two metric learning tasks with lack of supervision, i.e., unsupervised metric learning and semi-supervised metric learning.

4.1. Unsupervised Metric Learning

We consider the student embedding model trained by STML as our final model, which is evaluated and compared with state-of-the-art unsupervised metric learning methods and representative self-supervised learning methods.

4.1.1 Experimental Setup

Datasets and evaluation. The models are evaluated and compared on three benchmark datasets, i.e., CUB-200-2011 (CUB) [54], Cars-196 (Cars) [32], and Stanford Online Product (SOP) [48]; for their train-test splits, we directly follow the standard protocol presented in [60]. Performance on these datasets is evaluated by Recall@$k$, the fraction of queries that have at least one relevant sample in their $k$-nearest neighbors on a learned embedding space.

Embedding networks. For comparisons to previous work, we employ GoogleNet [50] and Inception-BN [24] as our teacher and student embedding model architecture. The dimension of embedding layers $g_t$ and $g_s$ are set as the output size of the last pooling, and that of $f_s$ follows each experimental setting. We append an $l_2$ normalization layer on top of the teacher so that the Euclidean distances in Eq. (5) are bounded for computing semantic similarities stably.

Implementation details. The student model of STML is optimized by AdamP with Nesterov momentum [22] for 90 epochs on a single Titan RTX; the learning rate is initialized to $10^{-4}$ and scaled down by the cosine decay function [36]. Training images are randomly cropped to $227 \times 227$ and flipped horizontally at random while test images are resized to $256 \times 256$ then center cropped. For all experiments, $\sigma$ in Eq. (1) is set to 3. The margin $\delta$, momentum coefficient $\mu$ and the number of nearest neighbors $k$ are 1.0, 0.999 and 10 on the CUB and Cars datasets, and 0.9, 0.9 and 4 on the SOP dataset. Note that $k$ is the nearest neighbor batch construction and that in Eq. (6) are identical.

Batch construction. Nearest neighbors used to construct mini-batches in STML are computed on the student embedding space and update at every epoch. On the CUB and Cars datasets, each mini-batch consists of 120 images including 24 queries and their 4 nearest neighbors. Since the SOP dataset has a tiny number of samples per class, we construct a mini-batch with 60 queries and their top-1 neighbors on the dataset. Also, all images are randomly augmented twice following the multi-view augmentation strategy in [28].

4.1.2 Quantitative Results

For fair comparisons to previous work, performance evaluation is done in the following settings: GoogleNet with 64/128/512 embedding dimensions, and Inception with BatchNorm with 512 embedding dimensions. All backbone networks in this experiment are pre-trained on ImageNet. The results on the three datasets are summarized in Table 1.
### Table 1. Performance of the unsupervised and supervised metric learning methods on the three datasets. Their network architectures are denoted by abbreviations, G–GoogleNet [50], BN–Inception with BatchNorm [24], where superscripts denote their embedding dimensions.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Arch.</th>
<th>CUB R@1</th>
<th>CUB R@2</th>
<th>CUB R@4</th>
<th>Cars R@1</th>
<th>Cars R@2</th>
<th>Cars R@4</th>
<th>SOP R@1</th>
<th>SOP R@10</th>
<th>SOP R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABIER [39]</td>
<td>G^{512}</td>
<td>57.5</td>
<td>69.7</td>
<td>78.3</td>
<td>82.0</td>
<td>89.0</td>
<td>93.2</td>
<td>74.2</td>
<td>86.9</td>
<td>94.0</td>
</tr>
<tr>
<td>ABE [30]</td>
<td>G^{512}</td>
<td>63.0</td>
<td>74.5</td>
<td>83.3</td>
<td>85.2</td>
<td>90.5</td>
<td>94.0</td>
<td>76.3</td>
<td>88.4</td>
<td>94.8</td>
</tr>
<tr>
<td>MS [53]</td>
<td>BN^{512}</td>
<td>65.7</td>
<td>77.0</td>
<td>86.3</td>
<td>84.1</td>
<td>90.4</td>
<td>94.0</td>
<td>78.2</td>
<td>90.5</td>
<td>96.0</td>
</tr>
<tr>
<td>PA [27]</td>
<td>BN^{512}</td>
<td>69.1</td>
<td>78.9</td>
<td>86.1</td>
<td>86.4</td>
<td>91.9</td>
<td>95.0</td>
<td>79.2</td>
<td>90.7</td>
<td>96.2</td>
</tr>
</tbody>
</table>

#### Supervised

<table>
<thead>
<tr>
<th>Methods</th>
<th>Arch.</th>
<th>SOP R@1</th>
<th>SOP R@10</th>
<th>SOP R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABIER</td>
<td>G^{512}</td>
<td>57.5</td>
<td>69.7</td>
<td>78.3</td>
</tr>
<tr>
<td>ABE</td>
<td>G^{512}</td>
<td>63.0</td>
<td>74.5</td>
<td>83.3</td>
</tr>
<tr>
<td>MS</td>
<td>BN^{512}</td>
<td>65.7</td>
<td>77.0</td>
<td>86.3</td>
</tr>
<tr>
<td>PA</td>
<td>BN^{512}</td>
<td>69.1</td>
<td>78.9</td>
<td>86.1</td>
</tr>
</tbody>
</table>

#### Unsupervised (Using ImageNet pre-trained embedding networks)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Arch.</th>
<th>SOP R@1</th>
<th>SOP R@10</th>
<th>SOP R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOM</td>
<td>G^{512}</td>
<td>55.0</td>
<td>67.5</td>
<td>78.6</td>
</tr>
<tr>
<td>ROUL</td>
<td>G^{512}</td>
<td>52.0</td>
<td>64.0</td>
<td>74.5</td>
</tr>
<tr>
<td>STML</td>
<td>G^{512}</td>
<td>55.0</td>
<td>67.5</td>
<td>78.6</td>
</tr>
</tbody>
</table>

#### Table 2. Performance on the SOP dataset using ResNet18 without pre-trained weights.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SOP R@1</th>
<th>SOP R@10</th>
<th>SOP R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCE</td>
<td>34.4</td>
<td>49.0</td>
<td>65.2</td>
</tr>
<tr>
<td>ISIF</td>
<td>39.7</td>
<td>54.9</td>
<td>71.0</td>
</tr>
<tr>
<td>PSLR</td>
<td>42.3</td>
<td>57.7</td>
<td>72.5</td>
</tr>
<tr>
<td>ROUL</td>
<td>45.4</td>
<td>60.5</td>
<td>74.8</td>
</tr>
<tr>
<td>SAN</td>
<td>46.3</td>
<td>61.9</td>
<td>77.0</td>
</tr>
<tr>
<td>STML</td>
<td>60.7</td>
<td>74.8</td>
<td>85.2</td>
</tr>
</tbody>
</table>

On all the datasets, our model achieves state-of-the-art performance in every setting. It improves the previous best score by a large margin, from 3.8% to 7.1% at Recall@1 with the 128 embedding dimensions. We note that our model clearly surpasses all existing models whose embedding dimension is 4 times higher on all the datasets, sometimes at 8 times higher. Moreover, on the CUB dataset, it even outperforms some of the supervised learning methods, e.g., ABIER [39] and MS [53], given the same embedding networks. These results suggest that our framework allows the model to learn compact yet effective embedding spaces by providing high-quality synthetic supervision. Last but not least, STML enables more efficient learning than existing pseudo labeling methods due to no use of off-the-shelf algorithms, especially on large-scale SOP dataset.

### 4.1.3 Unsupervised Metric Learning from Scratch

We demonstrate that STML is also effective for metric learning from scratch without pre-training on ImageNet. Following [60], we employ randomly initialized ResNet18 [21] with 128 embedding dimensions as the backbone network, and evaluate our model trained on the SOP dataset. As shown in Table 2, our model substantially outperforms existing methods including those for self-supervised representation learning in this setting. This result suggests that STML can generate reliable pseudo labels even on a randomly initialized embedding space, and our model trained by STML well generalizes to unseen class.

### 4.1.4 In-depth Analysis

#### Correlation between similarities and class-equivalence.

To justify the contextualized semantic similarity, we compare our three similarity metrics (i.e., pairwise similarity \( w_{ij} \), contextual similarity \( w_{ij}^{C} \), contextualized semantic sim-
Table 3. Ablation study of the components in STML on the CUB and Cars datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>STML</td>
<td>60.6</td>
<td>50.5</td>
</tr>
<tr>
<td>w/o contextual pairwise similarity</td>
<td>44.2</td>
<td>40.2</td>
</tr>
<tr>
<td>w/o pairwise similarity</td>
<td>56.0</td>
<td>45.0</td>
</tr>
<tr>
<td>w/o relaxed contrastive loss</td>
<td>53.4</td>
<td>38.7</td>
</tr>
<tr>
<td>w/o momentum update</td>
<td>56.8</td>
<td>46.1</td>
</tr>
<tr>
<td>w/o NN batch construction</td>
<td>54.6</td>
<td>44.8</td>
</tr>
<tr>
<td>w/o KL divergence</td>
<td>59.2</td>
<td>49.4</td>
</tr>
</tbody>
</table>

Figure 4. Recall@1 versus the hyper-parameters \(k\) and \(\sigma\) on the CUB and Cars datasets. Larger markers indicate our default experimental setting.

Figure 3. Comparison between the contextualized semantic similarity and its two components in terms of class-equivalence prediction accuracy on the CUB dataset with pre-trained weights (left) and SOP dataset without pre-trained weights (right).

Effect of hyper-parameters. We investigate how much sensitive STML is to the hyper-parameters \(k\) and \(\sigma\). Specifically, we measure performance of STML employing GoogleNet with 512 embedding dimensions on the CUB and Cars datasets while varying values of \(k\) and \(\sigma\). The results shown in Fig. 4 suggest that the performance of STML is consistently high when \(k < 14\), and \(\sigma\) has virtually no effect on the performance if \(\sigma > 1\) on both datasets. Moreover, whenever \(k < 14\) our model outperforms all existing unsupervised metric learning methods. Note that our final results reported in Table 1 are not the best scores in Fig. 4 since we did not tune the hyper-parameters on the test splits.

Comparison with recent SSL methods. STML has components used in recent self-supervised representation learning (SSL) methods such as the momentum encoder. To validate that the improvements of our framework do not originate from its structure, such methods are compared with STML under the same setting. Table 4 shows that MoCo and BYOL clearly underperform our method in all settings and achieve similar performance to unsupervised metric learning methods based on instance-level surrogate classes. Although they are structurally similar to STML, the reason for the degradation of performance is that they cannot capture intra-class variation unlike our method, and are

<table>
<thead>
<tr>
<th>Methods</th>
<th>Arch.</th>
<th>CUB</th>
<th>Cars</th>
<th>SOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCo [20]</td>
<td>G(_{128})</td>
<td>48.3</td>
<td>37.2</td>
<td>53.3</td>
</tr>
<tr>
<td>BYOL [18]</td>
<td>G(_{128})</td>
<td>47.7</td>
<td>31.5</td>
<td>48.7</td>
</tr>
<tr>
<td>MSF [31]</td>
<td>G(_{128})</td>
<td>46.0</td>
<td>30.5</td>
<td>41.6</td>
</tr>
<tr>
<td>STML</td>
<td>G(_{128})</td>
<td>59.7</td>
<td>49.0</td>
<td>65.8</td>
</tr>
<tr>
<td>MoCo [20]</td>
<td>G(_{512})</td>
<td>51.0</td>
<td>39.0</td>
<td>53.4</td>
</tr>
<tr>
<td>BYOL [18]</td>
<td>G(_{512})</td>
<td>50.9</td>
<td>38.5</td>
<td>51.2</td>
</tr>
<tr>
<td>MSF [31]</td>
<td>G(_{512})</td>
<td>49.6</td>
<td>32.5</td>
<td>47.8</td>
</tr>
<tr>
<td>STML</td>
<td>G(_{512})</td>
<td>60.6</td>
<td>50.5</td>
<td>65.3</td>
</tr>
</tbody>
</table>
Table 5. Performance of the supervised and semi-supervised methods on the two datasets. Network architectures of the methods are all ResNet50 (R50) [21] and superscripts denote their embedding dimensions. Also, the column “Init.” indicates whether the models are pre-trained on ImageNet or by SwAV. Our model and SLADE are all fine-tuned with the MS loss.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Arch.</th>
<th>Init.</th>
<th>CUB</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MAP@R</td>
<td>RP</td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
<td>25.0</td>
<td>35.8</td>
</tr>
<tr>
<td>Contrastive [9]</td>
<td>R50\textsuperscript{512}</td>
<td>ImageNet</td>
<td>26.4</td>
<td>37.5</td>
</tr>
<tr>
<td>MS [53]</td>
<td>R50\textsuperscript{512}</td>
<td>ImageNet</td>
<td>28.3</td>
<td>39.1</td>
</tr>
<tr>
<td>Proxy-Anchor [27]</td>
<td>R50\textsuperscript{512}</td>
<td>ImageNet</td>
<td>29.3</td>
<td>39.8</td>
</tr>
<tr>
<td>Contrastive [9]</td>
<td>R50\textsuperscript{512}</td>
<td>SwAV [5]</td>
<td>29.2</td>
<td>40.2</td>
</tr>
<tr>
<td>MS [53]</td>
<td>R50\textsuperscript{512}</td>
<td>SwAV [5]</td>
<td>31.7</td>
<td>42.0</td>
</tr>
<tr>
<td>Proxy-Anchor [27]</td>
<td>R50\textsuperscript{512}</td>
<td>SwAV [5]</td>
<td>31.7</td>
<td>42.0</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td></td>
<td></td>
<td>30.9</td>
<td>41.9</td>
</tr>
<tr>
<td>SLADE (MS) [14]</td>
<td>R50\textsuperscript{512}</td>
<td>ImageNet</td>
<td>37.8</td>
<td>47.9</td>
</tr>
<tr>
<td>STML (MS)</td>
<td>R50\textsuperscript{512}</td>
<td>SwAV [5]</td>
<td>33.9</td>
<td>44.4</td>
</tr>
<tr>
<td>SLADE (MS) [14]</td>
<td>R50\textsuperscript{512}</td>
<td>SwAV [5]</td>
<td>35.2</td>
<td>45.3</td>
</tr>
</tbody>
</table>

4.2. Semi-supervised Metric Learning

Semi-supervised metric learning aims to improve performance of supervised metric learning by exploiting additional unlabeled data [14]. STML can be employed for this task naturally since it provides a way of using the unlabeled data for training by predicting their class-equivalence relations as synthetic supervision. In this section, we validate the efficacy of STML for semi-supervised metric learning and compare it with SLADE [14], the first and only existing method for the task, on the CUB and Cars datasets.

4.2.1 Experimental Setup

Datasets and evaluation. We directly adopt the evaluation protocol of SLADE. Specifically, we employ two combinations of labeled and unlabeled datasets, \{CUB [54], NABirds [51]\} and \{Cars [32], CompCars [58]\}, and three performance metrics, MAP@R, RP, and Recall@1.

Embedding networks. For comparisons to SLADE, we employ ResNet50 [21] with 512 embedding dimensions as the backbone network of the teacher and student models. Before applying STML, these models are pre-trained either on ImageNet [12] or by SwAV [5], then fine-tuned with the MS loss [53] on a labeled dataset (i.e., CUB or Cars).

Implementation details. In this task, STML adopts an embedding model trained on a labeled dataset as the initial teacher and student models and trains them using both labeled and unlabeled data. Note that it computes and utilizes contextualized semantic similarities as supervision for labeled data as well as unlabeled data. For all experiments, \(m\) in Eq. (4) and \(\sigma\) in Eq. (10) are set to 0.9999 and 2, respectively, and the rest settings are the same as those for unsupervised metric learning.

4.2.2 Results

STML is compared with SLADE and state-of-the-art supervised metric learning methods on the two datasets. The results summarized in Table 5 show that STML improves performance of its supervised learning counterpart (i.e., MS) significantly regardless of the model initialization schemes. In particular, it enhances the performance by more than 5% in Recall@1 on both datasets for ImageNet pre-trained setting. Also, STML clearly outperforms SLADE in all the settings except one, where their gap is only 0.4% in RP.

5. Conclusion

We have proposed STML, a novel end-to-end unsupervised metric learning framework that estimates and exploits semantic relations between samples as pseudo labels. The pseudo labels in STML are estimated by investigating contexts of data as well as their pairwise distances on an embedding space. STML has achieved the state of the art on the three benchmark datasets for unsupervised metric learning, and demonstrated impressive performance even without pre-trained weights. Moreover, it has been applied to semi-supervised learning, and outperformed state of the art on the two benchmark datasets. Unfortunately, like other approaches, STML has a limitation in that it uses ImageNet pre-trained weights in some settings. In the future, we will explore extensions of our method where class labels are never involved in the training.

Acknowledgement. This work was supported by the NRF grant, the IITP grant, and R&D program for Advanced Integrated-intelligence for IDentification, funded by Ministry of Science and ICT, Korea (NRF-2021R1A2C3012728, NRF-2018R1A5A1060031, NRF-2018M3E3A1057306, No.2019-0-01906 Artificial Intelligence Graduate School Program–POSTECH).
References


